

Requested Patent: JP9034754A

Title:

INSTRUMENT AND METHOD FOR MEASURING PERFORMANCE  
OF PROGRAM ;

Abstracted Patent: JP9034754 ;

Publication Date: 1997-02-07 ;

Inventor(s): TANAKA TAKEHIKO ;

Applicant(s): FUJITSU LTD ;

Application Number: JP19950182623 19950719 ;

Priority Number(s): ;

IPC Classification: G06F11/34 ;

Equivalents:

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a means, with which measurement can be executed without preparing any driver programs dedicated to each route by facilitating environment preparation for performance measurement and the performance of various kinds of items can be measured simultaneously with time measurement as well, concerning a device for measuring the execution performance of a program. SOLUTION: A performance measuring part 200, in which the program is executed in a debugging mode for generating the trap each time an instruction is executed and control is assigned by the trap, is provided with an instruction execution time collecting means 6 for measuring the instruction execution time for each instruction, program execution time measuring means 9 for measuring the execution time of the program from this instruction execution time, measuring means 7 and 8 for a dynamic step and route trace to be simultaneously measured, execution control means 1 for designating a measurement start point and an end point and state operating means 2 for operating data and deciding any route.

特開平9-34754

(43)公開日 平成9年(1997)2月7日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 11/34

識別記号

庁内整理番号

7313-5B

F I

G 0 6 F 11/34

技術表示箇所

M

審査請求 未請求 請求項の数 8 O L (全 12 頁)

(21)出願番号 特願平7-182623

(22)出願日 平成7年(1995)7月19日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72)発明者 田中 岳彦

愛知県名古屋市東区葵1丁目16番38号 株  
式会社富士通愛知エンジニアリング内

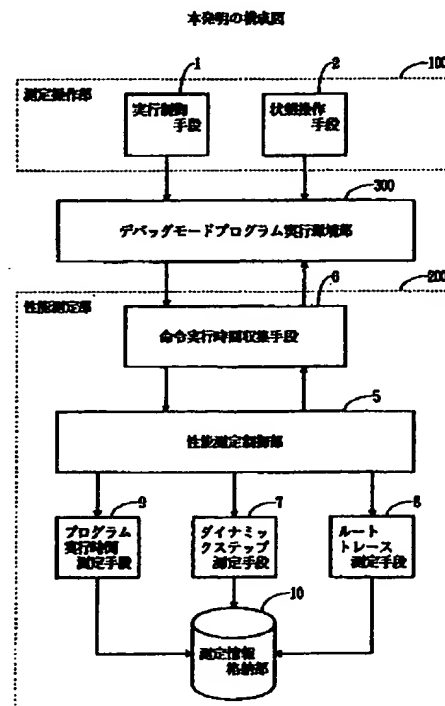
(74)代理人 弁理士 井桁 貞一

## (54)【発明の名称】 プログラムの性能測定装置と方法

## (57)【要約】

【目的】 プログラムの実行性能を測定する装置に関し、性能測定における環境準備を容易にし、各ルート専用のドライバプログラムを作成せずに測定を実施でき、各種項目も時間測定と同時に性能測定できる手段を提供することを目的とする。

【構成】 命令実行毎に割り出しを発生するデバッグモードでプログラムを実行し、割り出しにより制御を移された性能測定部200に、1命令毎の命令実行時間を計時する命令実行時間収集手段6と、これからプログラムの実行時間を測定するプログラム実行時間測定手段9と、同時に測定するダイナミックステップ、ルートトレースの測定手段7、8と、測定開始点、終了点を指定する実行制御手段1、データを操作し、ルートを決定する状態操作手段2とを備えるように構成する。



**【特許請求の範囲】**

【請求項1】 プログラムの命令の実行毎に割り出しを発生するデバッグモードを有するコンピュータのプログラムの性能測定装置であって、

測定対象プログラムを上記デバッグモードで実行し、上記発生する割り出しにより制御を移される性能測定部が割り出し元に制御を戻すとき計時を開始し、上記測定対象プログラムの1命令の実行後発生する割り出しにより制御が移されるとき計時を終了して上記1命令の実行時間として一時記憶する命令実行時間収集手段と、

上記一時記憶した1命令の実行時間を上記測定対象プログラムの測定開始から測定終了までに実行する命令について積算して記憶するプログラム実行時間測定手段と、を備えることを特徴とするプログラムの性能測定装置。

【請求項2】 上記割り出しにより制御を移される度に回数を計数して記憶するダイナミックステップ測定手段を追加して備えることを特徴とする請求項1のプログラムの性能測定装置。

【請求項3】 上記割り出しにより制御を移される度に測定対象プログラムの実行する命令のアドレスおよびコードを時系列で記憶するルートトレース測定手段を追加して備えることを特徴とする請求項1あるいは請求項2のプログラムの性能測定装置。

【請求項4】 上記測定対象プログラムの測定を開始する命令と測定を終了する命令をアドレスで指定し、上記測定対象プログラムを起動し、上記起動されたプログラムが上記指定した測定開始命令の実行に到ったとき上記プログラムの実行を中断し、上記中断したプログラムの実行を再開する実行制御手段と、

上記測定対象プログラムが実行を中断したとき、上記中断したプログラムの実行ルートを決めるデータの設定を行う状態操作手段と、を追加して備えることを特徴とする請求項1、請求項2あるいは請求項3のプログラムの性能測定装置。

【請求項5】 プログラムの命令の実行毎に割り出しを発生するデバッグモードを有するコンピュータのプログラム性能測定方法であって、

測定対象プログラムを上記デバッグモードで実行し、上記発生する割り出しにより制御を移される性能測定部が割り出し元に制御を戻すとき計時を開始し、上記測定対象プログラムの1命令の実行後発生する割り出しにより制御が移されるとき計時を終了して命令実行時間とし、上記命令実行時間を上記測定対象プログラムの測定開始から測定終了までに実行する命令について積算し、上記測定対象プログラムの実行時間とすることを特徴とするプログラムの性能測定方法。

【請求項6】 上記割り出しにより制御を移される度に測定対象とする命令の数を計数することを追加することにより測定対象プログラムのダイナミックステップ数を測定することを特徴とする請求項5のプログラムの性能

測定方法。

【請求項7】 上記割り出しにより制御を移される度に測定対象プログラムの実行する命令のアドレスおよびコードを時系列で記憶することを追加することにより、ルートトレースを測定することを特徴とする請求項5あるいは請求項6のプログラムの性能測定方法。

【請求項8】 上記測定対象プログラムの測定を開始する命令と測定を終了する命令をアドレスで指定し、上記測定対象プログラムを起動し、上記起動されたプログラムが上記指定した測定開始の命令の実行に到ったとき上記プログラムの実行を中断し、上記中断したプログラムの実行ルートを決めるデータの設定を行い上記中断したプログラムの実行を再開するようにすることを特徴とする請求項5、請求項6あるいは請求項7のプログラムの性能測定方法。

**【発明の詳細な説明】****【0001】**

【産業上の利用分野】 本発明はプログラムの実行性能を測定する装置と方法に関する。業務用のアプリケーションプログラムでは、実運用を開始する前に、プログラムの実行性能を測定し設計通りの性能が得られているか検証する必要がある。しかし、プログラムの性能を測定するためには、性能測定用の特殊な環境を用意する必要があり、プログラムの性能の問題点を開発工程の早い時期に検出することができていない。

【0002】 このため、アプリケーションプログラムの性能測定を容易に実施できるようにする必要がある。

**【0003】**

【従来の技術】 従来のプログラムの性能測定においては、測定したいアプリケーションプログラムを性能測定用の環境下で起動し、プログラムの特定箇所を対象として性能測定を行っていた。この場合、測定用の環境下で動作するプログラムの特定のルートだけしか性能を測定できず、プログラム上の別ルートを測定するためには、そのルートを実行させるためのドライバプログラムなどの環境を準備する必要があった。このため、特殊なルートなどで頻繁に走行しないルートを測定するためには、各ルート毎のドライバプログラム環境を準備する必要があり、簡単に性能を測定することができなかった。

【0004】 時間測定の例では特開平3-102437において、コンパイラが作成したデバッグのシンボル情報を利用して、性能測定の対象となるプログラムにトラップ命令を埋め込むことで、性能測定の対象となるプログラムに対し性能測定ルーチンを呼び出すアプリアロセッサの実行や、再コンパイル、再リンクを不要とすることを特徴とした方式が示されている。

【0005】 しかし、プログラムの実行ルートを容易に設定する手段やダイナミックステップなどの各種の性能項目の測定も時間測定と同時に進められるような柔軟で操作の容易な性能測定を可能とするものとはなっていない。

## 【0006】

【発明が解決しようとする課題】上記のように、従来の技術では、性能測定したいアプリケーションプログラムを性能測定用の環境下で起動するので、プログラム上のあるルートを測定するためには、そのルート用のドライバプログラムを作成する必要がある、環境準備にかなりの労力が必要となる問題が生じていた。

【0007】本発明はこのような点にかんがみて、性能測定における環境準備を容易にし、各ルート専用のドライバプログラムを作成せずに測定を実施でき、また、各種の性能項目の測定も時間測定と同時に柔軟に測定ができる性能測定手段を提供することを目的とする。

## 【0008】

【課題を解決するための手段】上記の課題は下記の如くに構成されたプログラムの性能測定装置によって解決される。

【0009】図1は、本発明の構成図である。300はデバッグモードプログラム実行環境部であり、測定対象プログラムを1命令毎に割り出しを発生させるデバッグモードで実行する環境である（請求項1～8）。

【0010】100は測定操作部であり、上記測定対象プログラムの測定を開始する命令と測定を終了する命令をアドレスで指定し、上記プログラムを起動し、上記起動されたプログラムが上記指定した測定開始アドレスの命令の実行に到ったとき上記プログラムの実行を中断し、この中断したプログラムの実行を再開する実行制御手段1と、上記測定対象プログラムが実行を中断したとき、上記中断したプログラムの実行ルートを決定するデータの設定を行う状態操作手段2とからなっている（請求項4、8）。

【0011】200は性能測定部であり、上記測定対象プログラムをデバッグモードで実行するとき発生する割り出しにより制御を移され、1命令毎に必要な性能測定情報の処理を行い上記測定対象プログラムの割り出し元に制御を帰すものである。

【0012】これを構成する各手段については以下に示す（請求項1～3、5～7）。命令実行時間収集手段6は上記発生する割り出しにより制御を移される性能測定部が割り出し元に制御を戻すとき計時を開始し、上記測定対象プログラムの1命令の実行後発生する割り出しにより制御が移されるとき計時を終了して上記1命令の実行時間として一時記憶するものである（請求項1、5）。

【0013】性能測定制御部5は性能測定部200の各手段の動作の実行を制御するものであり、情報を一時記憶する不図示の記憶部を含むものである（請求項1～3、5～7）。

【0014】プログラム実行時間測定手段9は、上記一時記憶された命令実行時間を上記測定対象プログラムの測定開始から測定終了までに実行する命令について積算

して記憶するものである（請求項1、5）。

【0015】ダイナミックステップ測定手段7は上記割り出しにより制御を移される毎に回数を計数して記憶するものである（請求項2、6）。ルートトレース測定手段8は上記割り出しにより制御を移される毎に測定対象プログラムの実行する命令のアドレスおよびコードを時系列で記憶するものである（請求項3、7）。

## 【0016】

【作用】測定対象プログラムをデバッグモードプログラム実行環境部300でデバッグモードで実行することにより、発生する割り出しにより性能測定部200に制御が移され、1命令毎に必要な性能測定情報の処理を行い上記測定対象プログラムの割り出し元に制御を帰すようにプログラムの論理に従った命令の実行を追跡しながら性能測定を行うようにしている。

【0017】プログラムの実行時間の測定は上記発生する割り出しにより制御を移される命令実行時間収集手段6が割り出し元に制御を戻すとき計時を開始し、上記測定対象プログラムの1命令の実行後発生する割り出しにより制御が移されるとき計時を終了して上記1命令の命令実行時間として一時記憶し、これをプログラム実行時間測定手段9により測定開始から測定終了までに実行する命令について順次積算して記憶するようにしている（請求項1、5）。

【0018】ダイナミックステップの測定はダイナミックステップ測定手段7により上記割り出しで制御を移される毎に回数を累積して記憶するようにしている（請求項2、6）。

【0019】ルートトレース測定手段8では上記割り出しにより制御を移される毎に測定対象プログラムの実行する命令のアドレスおよびコードを時系列で蓄積記憶するようにして実行したプログラムの命令のルートを測定記録するようになっている（請求項3、7）。

【0020】100は測定操作部では、実行制御手段1により上記測定対象プログラムの測定を開始する命令のアドレスと測定を終了する命令のアドレスを指定し、上記プログラムを起動し、上記起動されたプログラムが上記指定した測定開始命令の実行に到ったとき上記プログラムの実行を中断し、状態操作手段2では上記中断したプログラムの実行ルートを決定するデータの設定を行い、上記作実行制御手段1により再度上記中断したプログラムの実行を再開するようになっている（請求項4、8）。

## 【0021】

【実施例】図2に本発明の実施例の構成図を示す。測定操作部100は測定対象プログラム300の測定条件、測定環境を操作するものであり、実行制御部1、状態操作部2とデータ解析部23を含む。なお、測定操作部100は従来技術であるデバッガの機能である。

【0022】実行操作部1は測定対象プログラム300

をデバッグモードで実行できる状態に設定し、性能測定を行う命令の先頭と測定を終了する命令をそれぞれ測定点XA25、測定点XB26としてそのアドレスを指定して設定する。アドレスはデバッグのシンボル情報を用いてシンボルで指定したり、オブジェクトコードのリストからアドレスを調べて指定することもできる。

【0023】実行ルートの指定は測定対象プログラムのデータ部27にあるデータを変更して設定することにより内部ロジックで決まるルートを変更できる。すなわち、状態操作部2によりデバッグの機能であるデータの操作を行い必要なデータの設定を行うことによりルートの指定ができる。

【0024】操作は測定開始時および上記設定した測定点に到達したときにオペレータに制御が与えられ、ディスプレイ画面、キーボード、マウスなどの入出力機能を有した操作端末21によりオペレータとの会話により行う。測定結果のデータについては上記操作端末21に表示するほか、解析結果を含めて出力装置22に記録することができるようにしている。

【0025】性能測定部200は測定対象プログラム300の実行にともない発生する性能情報を収集、記憶し、処理して測定データ格納部10に記録するものである。次に、図3に示す本発明の実施例の操作フローチャートを説明する。性能測定項目の組み合わせは種々あるがここではプログラムの実行時間の測定を例として説明する。

【0026】ステップS31では実行時間を測定しようとするプログラムを実行制御部1により指定し、プログラムを性能測定可能な環境下で実行可能な状態とする。つまり、プログラムを配置し、デバッグモードで実行可能とするもので、上記したデバッグの機能を利用する。

【0027】ステップS32において、上記設定したプログラム内の実行時間の測定を開始する位置として測定点XA25と終了する位置として測定点XB26を対応するプログラムリストにあるアドレスを指定して設定する。

【0028】ステップS33ではプログラムの実行を開始するように実行制御部1により指示する。プログラムは通常の起動と同様入口点から実行を開始する。すなわち、通常のプログラムの動作でのデータの初期設定等は測定対象プログラムに備わった機能がそのまま動作する。ただし、実行モードは上記設定されたようにデバッグモードとなっているので測定対象プログラム300は1命令実行毎に割り出しを発生するようになり、割り出しにより性能測定部200に制御が渡ることになる。そこで、アドレスが上記設定した測定開始すべきところすなわち測定点XA25に到達したか否かの検出を行う。

【0029】ステップS34に示すように測定点XA25に到達したことを実行制御部1からオペレータに通知され、操作端末21からオペレータが状態操作部2を操

作することが可能となる。

【0030】オペレータはステップS35においてプログラム環境の設定を変更することができる。ここで、状態操作部2を利用してプログラムのメモリの内容などの動作環境を設定し、プログラム上の意図した実行ルートが動作するように準備する。

【0031】また、上記設定した二つの測定点間の設定したルートを実行するときに測定すべき測定項目をメニューから選択する。メニューにはプログラム実行時間の測定の他にダイナミックステップ数や実行したプログラムのルートの測定があるがここでは実行時間の測定を指定する。

【0032】ステップS36において、性能測定をしながら、プログラムの実行を再開することを実行制御部1に指示する。プログラムの実行にともない各命令の実行の毎に制御が渡される性能測定部200では上記指定された性能測定項目に従い性能情報を収集して行く。ここではプログラムの実行時間の測定を行うために各命令ごとの実行時間を時刻情報を読み取りながらプログラムの実行に要した時間を測定して行く。同時に性能測定部200ではプログラムの実行アドレスが測定終了を示す測定点XB26に到達したか否かを監視する。

【0033】ステップS37において測定点XBに到達したときには実行制御部1からオペレータにその由通知され、オペレータの指示待ちの状態になる。ステップS38において、オペレータの判断により、データの設定を変えることにより別のルートの実行時間の測定を行うようにするか、あるいは測定を終了することにするのかを決めることになる。再度、測定を行うときは実行制御部によりステップS32に移行する。

【0034】性能測定部200の動作の説明に入る前に、図4によりデバッグモードにおける割り出しの説明を行う。測定対象プログラム300の図中には実行する順序に命令を並べてある。すなわち、物理的なアドレスの順ではなく分岐命令などで実行される順序が変わるときには分岐先の命令を次に並べるようにしたものである。

【0035】A命令実行41の後では矢印のように割り出しが発生し、測定対象プログラム300の命令の実行は次に進まず、割り出しにより性能測定部200に制御が渡る。性能測定部200での動作の詳細は図5、図6により後述するが、本来の処理である性能測定処理では測定内容により、あるいは測定データを外部記憶媒体に格納するような時間を要するI/O動作が入ったりすることにより性能測定部200の処理に要する時間が一定するものではなく、さらに測定対象プログラム300の命令の実行ステップ数に比較して多くのステップが必要となり、特に時間の測定においては何処で時計を記録するか注意が必要である。したがって、割り出し、復帰のためのレジスタなどの測定対象プログラム300の状態

の保存と復元を最小限に止め、時計の読み取りをできるだけ入口部と出口部に近いところで行うようにして時間の測定に供するようにしている。逆にこのようにしておけば性能測定処理45ではどのような処理を行っても性能測定の精度には影響を与えないようにできる。

【0036】性能測定部200から復帰したときは、上記したようにA命令41が実行された結果として次に実行されるべき命令B42に制御が移る。B命令42が実行されると上記と同様に割り出しにより性能測定部の先頭に制御が移る。そして、性能測定処理45が終了すると性能測定部200からC命令43に実行が戻り以下同様に測定対象プログラム300の命令を実行して行く。

【0037】このように、測定対象プログラム300の1命令実行の毎に割り出しにより、1命令毎に各種の性能測定を進めることができる。図5と図6には性能測定部の動作のフローチャート(その1、その2)を示す。また、図7には性能測定処理に使用する測定情報記憶テーブルの構成説明図を示してあり、これらを参照しながら動作を説明する。

【0038】ステップS501では次のステップS502で行う時計を一次記憶するために必要な最低限のレジスタの保存を行う。ステップS502において割り出しの発生する直前に実行した測定対象プログラムの命令の実行終了時刻として時計を読み取り命令実行終了時刻として測定情報記憶テーブル72のTGに記憶する。次にその他のレジスタ保存をステップS503で行い性能測定部の本体の動作に進む。

【0039】まず、測定情報記憶テーブル72のTBにある測定中フラグを調べる。測定中フラグは測定開始の測定点XA25と測定終りの測定点XB26の間の測定対象の部分の命令を実行しているのか否かを示すもので、測定対象部分での割り出し時にはONとしておくものである。

【0040】まだ、測定対象部分に入っていないときにはステップS508でアドレスが測定点XAに到達したか否かを判定する。測定点XAではないことがわかると性能測定部200の本体の動作はなにもせずに測定対象プログラム300に戻ることになるが、S511～S513により割り出し元への復帰の手順としてレジスタの復旧を行う必要がある。このように測定開始点に到達するまでは実質なにもせず測定開始点の検出のみを行っている。

【0041】ステップS508において測定開始点に到達したことを検出したときは、ステップS509に移り測定中フラグTBをONとする。これにより、次に割り出しにより性能測定部200に制御が渡ったときには性能測定のデータが収集されることになる。

【0042】ステップS510では制御が測定操作部100に移されオペレータに測定開始点に到達したことを知らせ、オペレータの指示の入力待ちとなる。そこで、

意図するルートを実行するプログラムが進むようにデータの確認を行い変更すべきデータの設定を行う。必要な設定を確認した後オペレータの指示で性能測定部200に再開のために制御が戻る。

【0043】ステップS511以下は上記説明したように測定対象プログラム300に戻る手順であるが、上記で説明を省略したステップS512における時計の読み取りは次に実行される測定対象プログラムの命令の実行開始時刻として測定情報記憶テーブル72のTHに記憶する。

【0044】次に、ステップS504に戻って割り出しが発生したときに測定中フラグTBがONとなっていた場合は性能測定中でありステップS505に移る。まず、プログラム実行時間の測定のための処理としてステップS505からステップS507までで各実行した命令の開始と終了時刻の差を積算する処理を行う。

【0045】ステップS505では第1次命令実行時間として今回の性能測定部200の先頭のステップS502で読み取った時刻、これは今回の割り出しの発生する直前に実行した命令の実行終了時刻であり、これと前回の性能測定部200の最後のステップS507で読み取った時刻、すなわち前回の割り出しの発生した後実行した命令の実行開始時間との差である。この第1次命令実行時間には実際に命令を実行した時間に加えて割り出しのための時間、割り出し元にもどる時間が含まれている。これらの時間は限られたレジスタの保存、復旧の時間など予め測定した固定の値の時間であり測定情報記憶テーブルのTKに割り出し処理時間という定数として記憶されている。

【0046】ステップS506では第2次命令実行時間として上記の補正を行ったものを算出しテーブルのTJに記憶している。ステップS507では測定中の間すなわち測定中フラグTBがONの間について第2次命令実行時間TJを積算して、測定中に実行した命令のトータルのプログラム実行時間としてテーブルのTLに記憶している。

【0047】ステップS601では測定項目としてダイナミックステップのカウントが必要か、測定条件を確認し、必要なときは測定情報記憶テーブルのTMにカウントするようになっている。

【0048】ルートトレースについても同様にステップS603で要否を判定して、必要ならステップS604とステップS605において測定情報記憶テーブルのTOにあるルートトレースレコード74に測定対象プログラム300の実行した命令のアドレス情報と命令コードを対にして累積して記憶する。記録するレコード74がいっぱいになったら測定データ格納部10に書き出して格納するようにしている。

【0049】測定対象項目の処理が終了したときステップS606においてアドレスが測定点XBと一致したと

きには測定終了とし、測定を続行するときにはステップS511から測定対象プログラム300に戻る処理を行う。

【0050】測定終了の処理はステップS607で今回の測定区間における測定データを測定データ格納部10に記録するべく出力し、後刻、データ解析部23で処理加工が行える様にしておく。

【0051】ステップS608で測定中フラグをOFFとして、ステップS609で測定操作部100に制御を移行し、オペレータの指示による新たな測定点の設定を行う。このとき、新たな測定のための測定情報記憶テーブル72の初期設定が行われる。ステップS511に移りレジスタの復旧を行って新たに測定対象プログラムの実行に戻る。

【0052】以上のようにして任意の区間について設定したデータによって定まるルートを測定対象プログラム300を実行させながら指定した性能項目のデータを測定することができる。

【その他の実施例】上記実施例においては性能測定項目として、プログラム実行時間、ダイナミックステップ数、ルートトレースを例にあげたがこれ以外に、1命令毎に状態を監視し、指定した状態に基づいて性能情報を記憶したり、情報処理を行うような任意の性能測定を行うようにしてもよい。

【0053】1命令の実行時間の測定を割り出し元に戻る時刻と割り出しの時刻を時計を読み取る様に行っているが、これをタイマー等他の計時手段を用いてもよいし、割り出し、復帰に伴う処理時間の補正を1命令毎に行っているが、これを測定した命令数を用いて一括して補正してもよい。

【0054】また、測定操作部で測定範囲の指定やデータの設定を操作端末からのオペレータの操作によって行うようにしているが、従来技術として行われているように、オペレータにより与えるデータの操作やコマンド操

作を自動的にシステムで発生させ、性能測定の連続自動測定を行うようにしてもよい。

【0055】

【発明の効果】以上の説明から明らかなように本発明によればプログラムの性能測定における環境準備を容易にし、各実行ルート専用のドライバプログラムを作成せずに測定を実施でき、また、各種の性能項目についても時間測定と同時に柔軟に測定ができる性能測定手段を提供し、コンピュータを利用した情報処理システムの性能向上と信頼性向上、さらにはシステム構築の生産性向上等、という著しい工業的效果がある。

【図面の簡単な説明】

【図1】 本発明の構成図

【図2】 本発明の実施例の構成図

【図3】 本発明の実施例の操作フローチャート

【図4】 デバッグモードにおける割り出しの説明図

【図5】 性能測定部の動作のフローチャート（その1）

【図6】 性能測定部の動作のフローチャート（その2）

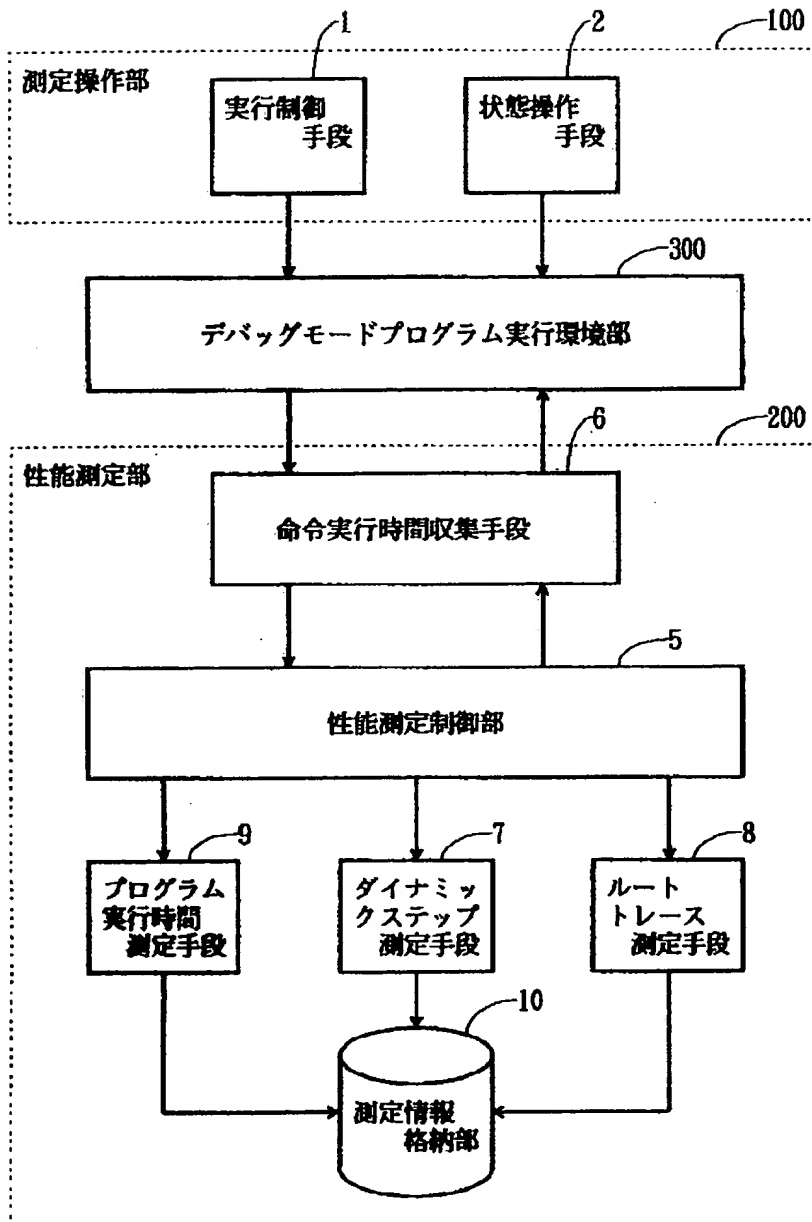
【図7】 測定情報記憶テーブルの構成説明図

【符号の説明】

- 1 実行制御手段
- 2 状態操作手段
- 5 性能測定制御部
- 6 命令実行時間収集手段
- 7 ダイナミックステップ測定手段
- 8 ルートトレース測定手段
- 9 プログラム実行時間測定手段
- 10 測定情報格納部
- 100 測定操作部
- 200 性能測定部
- 300 デバッグモードプログラム実行環境部

【図1】

## 本発明の構成図



【図7】

## 測定情報記憶テーブルの構成説明図

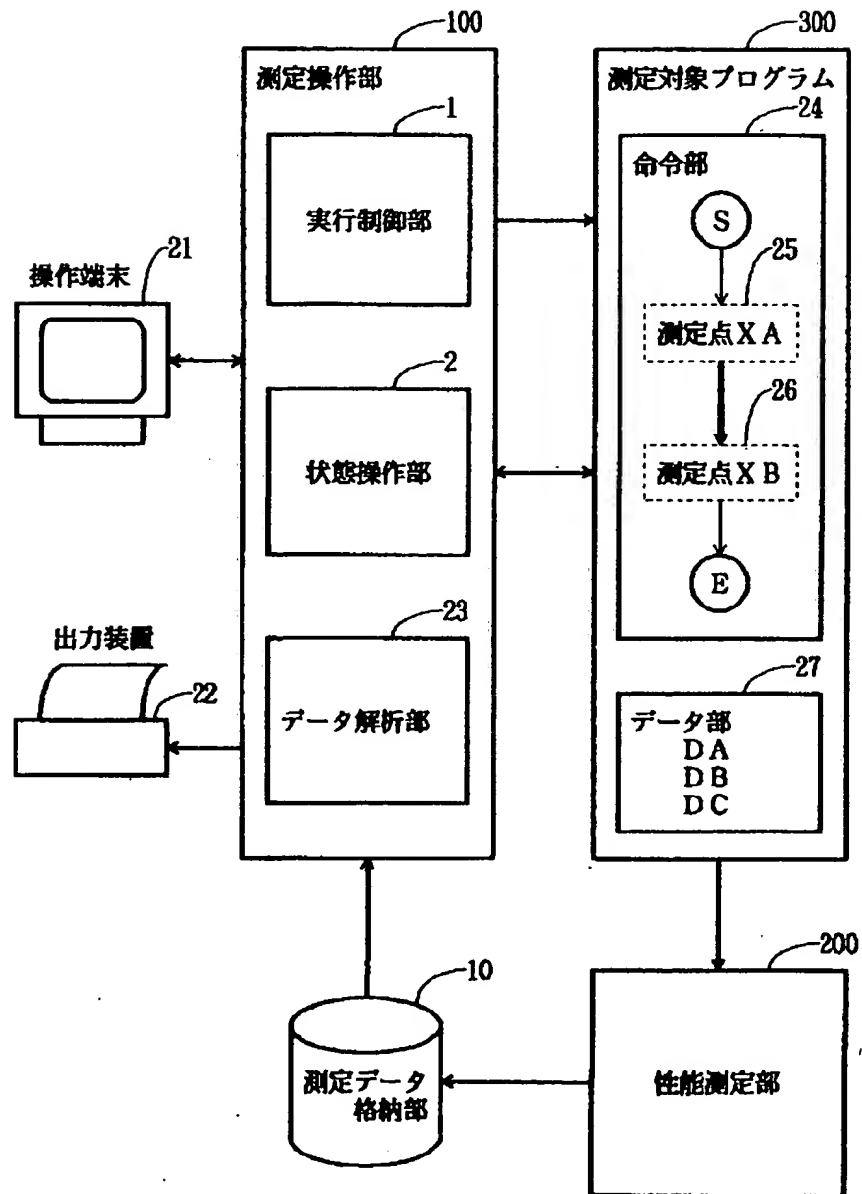
番号		内容
TA	測定番号	
TB	測定中フラグ	
TC	レジスタ保存1	
TD	レジスタ保存2	
TE	測定点XAアドレス	
TF	測定点XBアドレス	
TG	命令実行終了時刻	
TH	命令実行開始時刻	
TI	第1次命令実行時間	
TJ	第2次命令実行時間	
TK	割り出し処理時間	
TL	プログラム実行時間	
TM	ダイナミックステップ数	
TN	トレースレコードポインタ	
TO	トレースレコード	

アドレス	命令
AA	IA
AB	IB
AC	IC

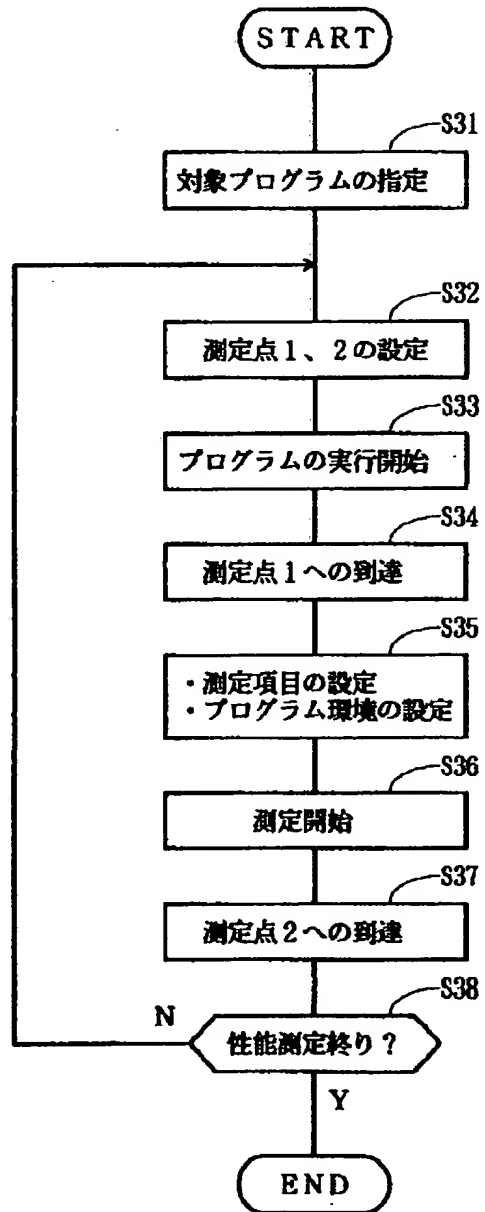
【図2】

本発明の実施例の構成図



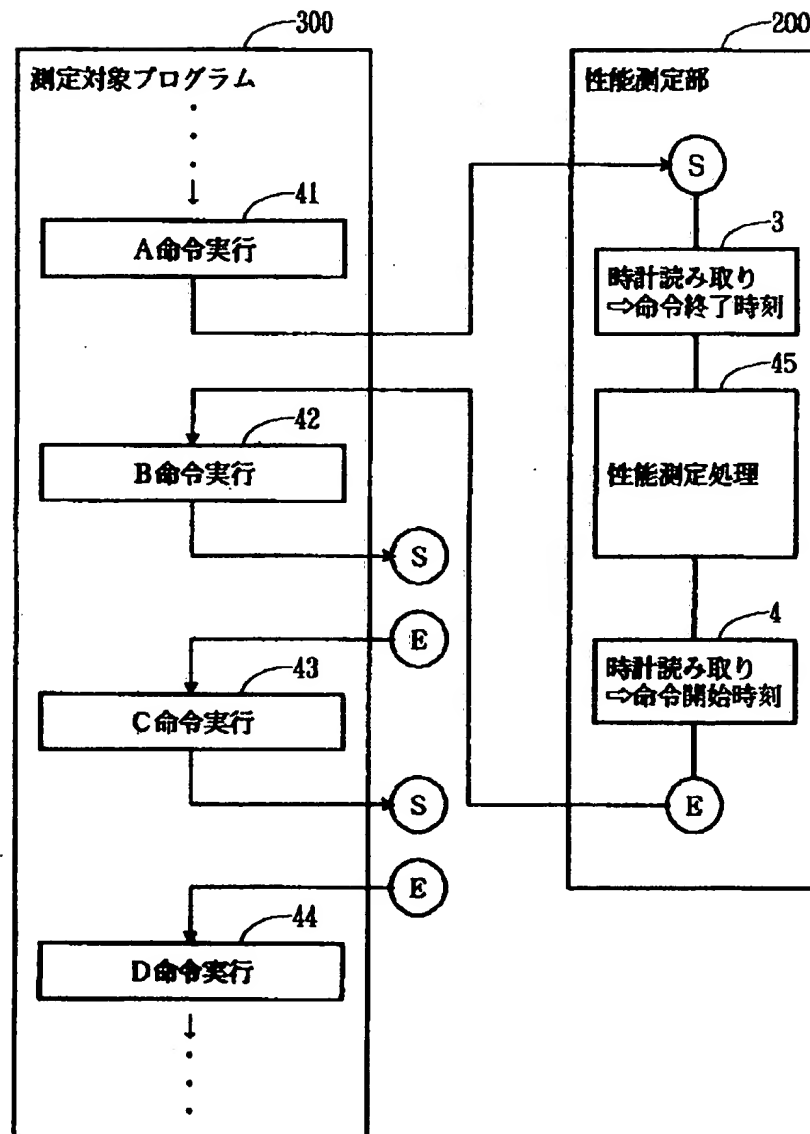
【図3】

本発明の実施例の操作フローチャート



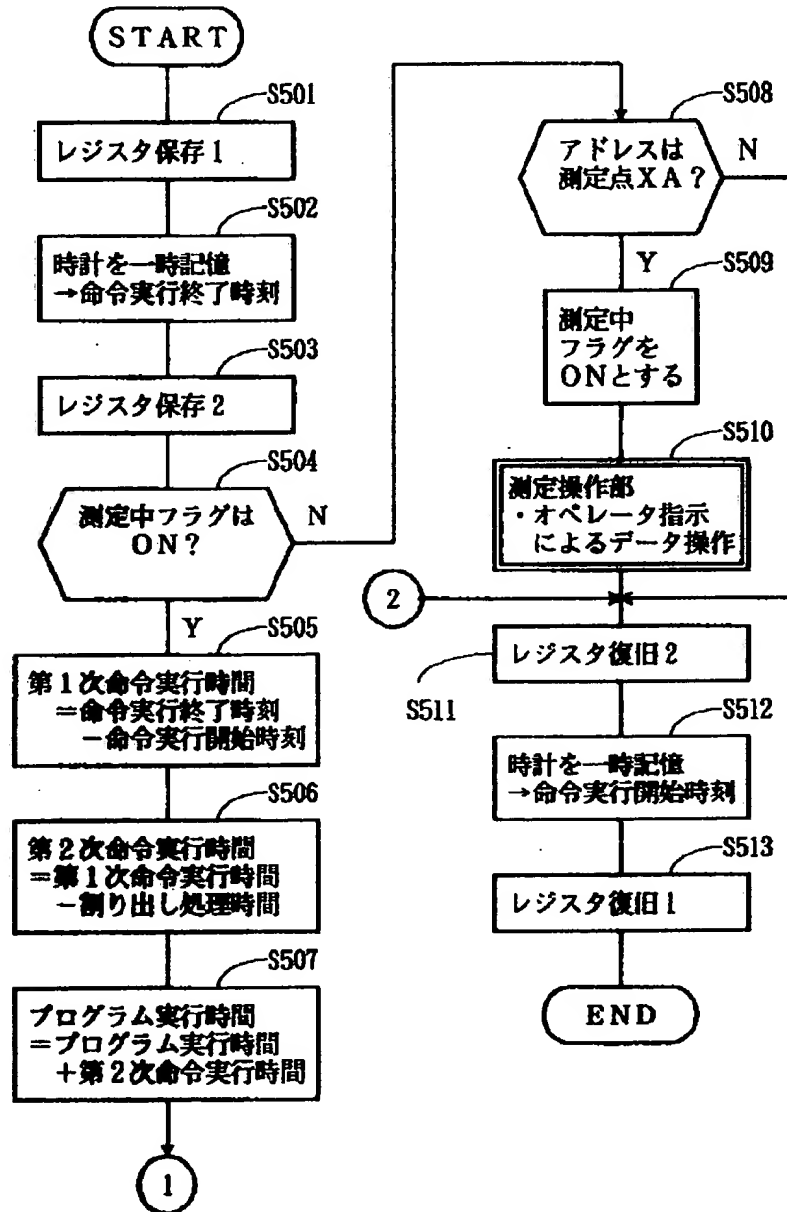
【図4】

デバッグモードにおける割り出しの説明図



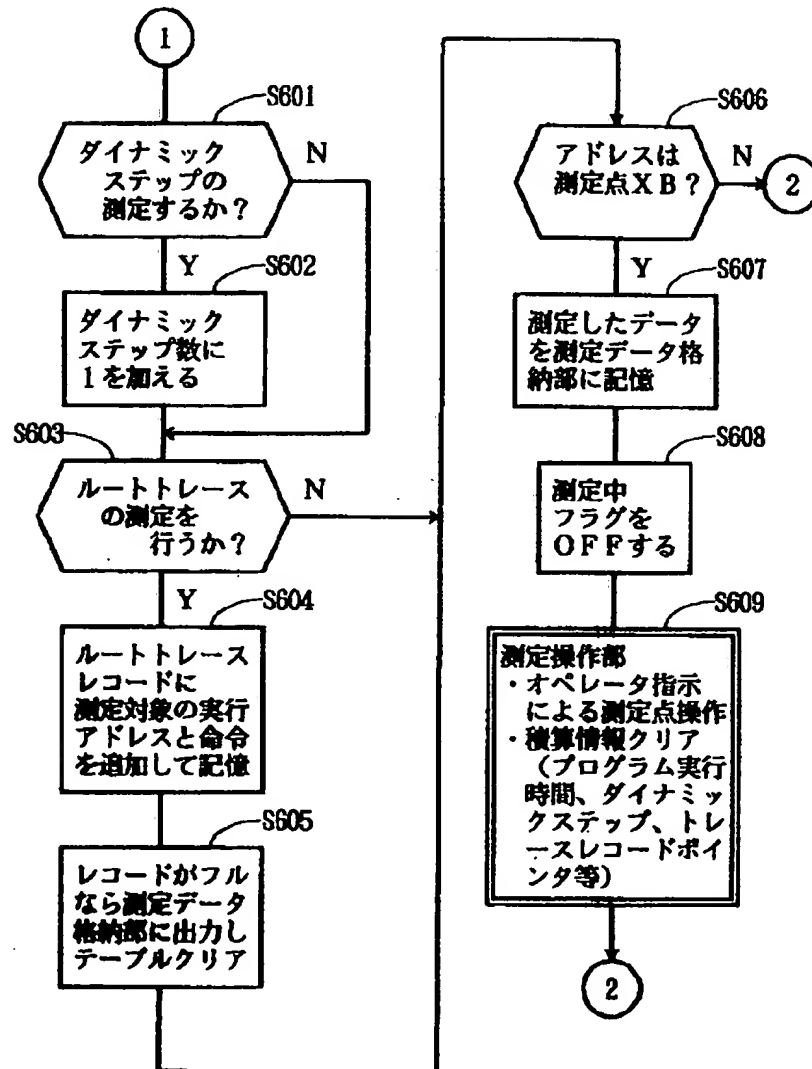
【図5】

## 性能測定部の動作のフローチャート（その1）



【図6】

## 性能測定部の動作のフローチャート（その2）



*This Page Blank (uspto)*